**David Chappell**

June 2011

# CLUSTER COMPUTING TODAY

## WHAT'S CHANGED AND WHY IT MATTERS

**Sponsored by Microsoft Corporation**

Copyright © 2011 Chappell & Associates

One way to make an application run faster is to divide its work into multiple parts, then run those parts simultaneously on a group of computers. Known as *cluster computing*, this approach has long been used with scientific and technical applications. Today, however, changes in how clusters can be built and in the kinds of applications they can run are making this idea useful for a wider set of problems across a broader range of organizations.

Traditional cluster-based applications are still important, of course. In fact, more and more scientific and technical problems can be addressed with software running on clusters. But compute clusters are also used today for financial applications, for applications that process very large amounts of data, and for other problems. Just as important, the barriers to entry for using a cluster have become much lower. The technology is now accessible even to small and mid-size organizations.

This short description examines these changes through the lens of Windows HPC Server, Microsoft's technology for cluster computing. With clusters now both more useful and more approachable, every organization should take another look at this technology.

## What's Changed: A Modern View of Cluster Computing

Traditionally, a compute cluster consisted of a group of server machines installed in an organization's data center. Creating a cluster meant buying and managing a significant number of these servers, then trying to keep them busy to justify the investment. This made sense only for organizations with a substantial ongoing requirement to run applications on a cluster.

There are more options today. For example, a cluster created using Windows HPC Server 2008 R2 can contain any combination of the following:

- On-premises servers, as in traditional compute clusters.
- Desktop workstations, which can become part of a cluster when they're not being used. Think of a financial services firm, for instance, which probably has many high-powered workstations that sit idle overnight.
- Cloud instances provided by public cloud platforms. These instances can be created on demand, used as long as needed, then shut down.

The ability to create compute clusters using desktop workstations and cloud instances is an important advance. Adding desktops to a cluster essentially provides free resources, since the machines otherwise aren't being used. Adding cloud instances isn't free, but because instances can be created and shut down on demand, the user pays only for the time they're being used. Both of these options allow creating clusters with a significantly smaller up-front investment than before. Using cluster computing no longer requires buying racks full of on-premises servers.

Just as the kinds of nodes a cluster can contain have broadened, so have the kinds of applications a cluster can run. For instance, Windows HPC Server 2008 R2 Service Pack (SP) 2 today supports several application types, including the following:

- MPI applications, where the components in a running application interact via the Message Passing Interface (MPI). This category includes many of the scientific and technical applications that have traditionally been classed as high-performance computing (HPC), including fluid dynamics, car crash simulations, and many more.

- Embarrassingly parallel applications, where the components in a running application don't interact with one another. (The name is used because making this kind of application run in parallel is almost embarrassingly easy.) An embarrassingly parallel application might consist of separate executables or be divided into a group of services invoked by a client while the application runs. In Windows HPC Server, this latter option is known as a *service-oriented architecture (SOA)* application. One common use of embarrassingly parallel applications is to do financial modeling, such as estimating the risk of an investment portfolio.

- Excel applications, where calculations done by an Excel workbook can be offloaded onto a cluster. Work that takes hours on a single desktop workstation might instead be completed in minutes using a cluster.

- Data-intensive applications, which need to work with large amounts of unstructured, non-relational data. More and more organizations face these kinds of problems today with customer data, Web logs, information from sensors, and more. Using a compute cluster can sometimes be the best option for storing and processing this information.

Supporting a broader set of applications makes clusters more useful. Making it easier to create clusters lets them be accessible to more businesses and public sector organizations. Taken together, these advances mean that cluster computing is now relevant to many more people than it was just a few years ago.

Windows HPC Server provides a concrete example of these changes. Figure 1 shows the product's evolution since 2006.

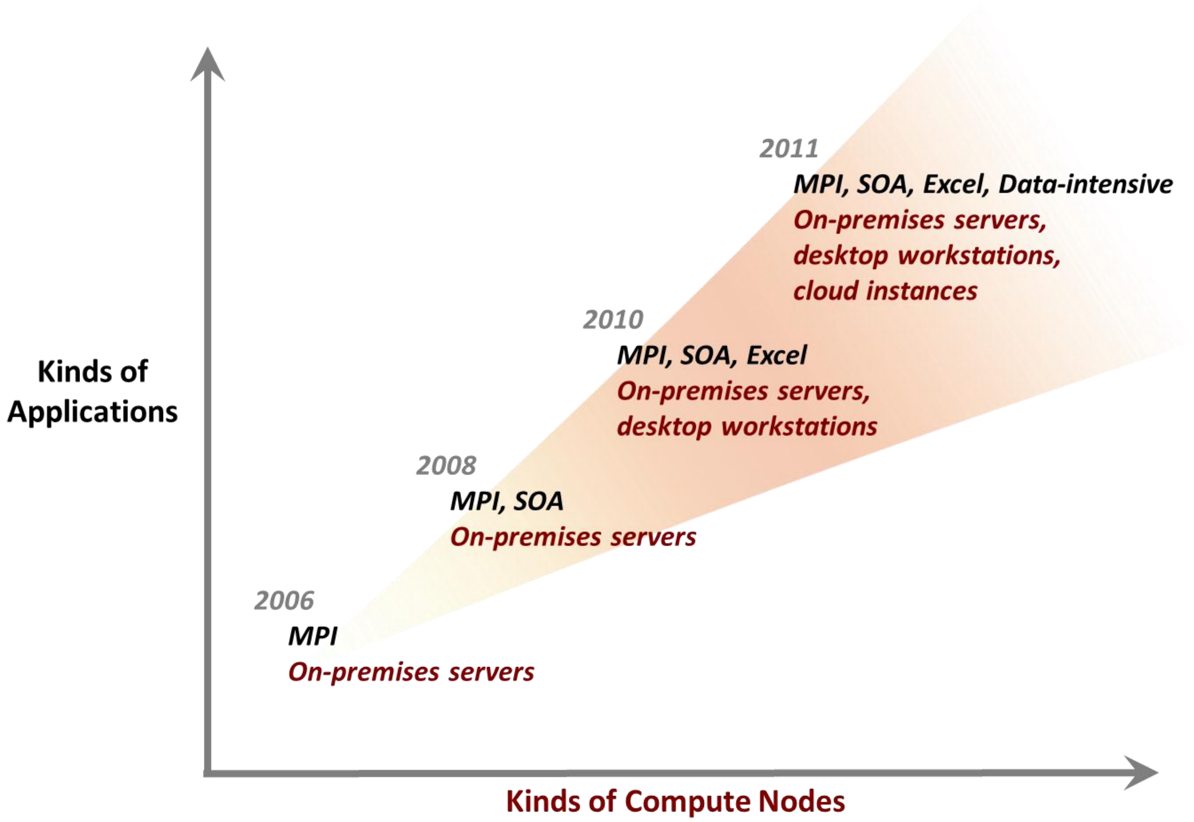


**Figure 1: How clusters can be created and used has expanded, as the changes in Windows HPC Server illustrate.**

In 2006, Windows HPC Server ran solely on clusters built using on-premises servers, and it supported only MPI applications. In other words, it was a traditional clustering technology focused on scientific and technical problems. Support for SOA applications was added in 2008, with the 2010 release adding the ability to run MPI, SOA, and Excel applications on clusters built from on-premises servers and desktop workstations. The product's 2011 release brought even more change, adding support for data-intensive applications and clusters that contain cloud instances.

This evolution provides a clear illustration of the changes in cluster computing. Starting from the specialized world of MPI applications, Windows HPC Server today addresses a significantly broader range of problems. And rather than require a large initial investment in on-premises servers, the technology now presents a lower barrier to entry. Once applicable to a quite narrowly focused world, cluster computing today is relevant to a much larger set of organizations.

## Why It Matters: Example Uses of Clusters Today

To get a better sense of what these changes mean, it's useful to look at specific scenarios. What follows describes two Windows HPC Server examples: using an on-demand cluster and creating data-intensive applications. Both demonstrate ways to use cluster computing that weren't possible just a few years ago, and both provide good examples of how the changes just described make this technology more useful.

## Using an On-Demand Cluster

In Windows HPC Server, a cluster contains some number of *compute nodes*, all of which are managed by another computer called the *head node*. The head node must be an on-premises server today, but it's entirely possible to create a cluster where all of the compute nodes are cloud instances. Figure 2 shows how this looks.

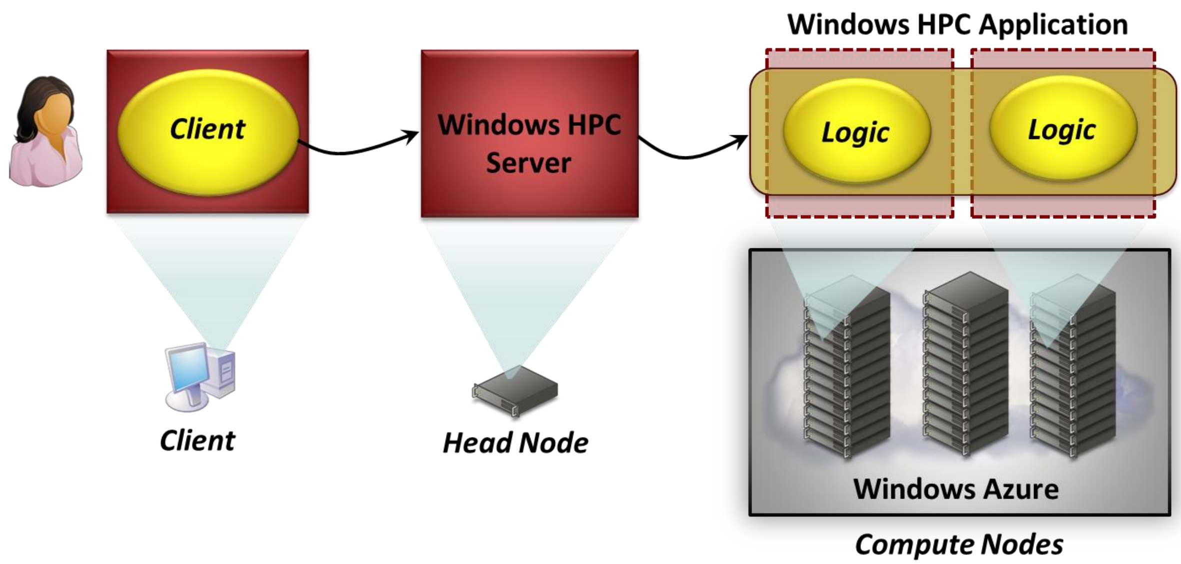


**Figure 2: An on-demand cluster requires only a single computer—the head node—on-premises, with all of its compute nodes running in the cloud.**

As the figure shows, a client submits an application (also referred to as a *job*) to the head node, which then starts the application running on the cluster's compute nodes. Here, each chunk of the application's logic runs in a cloud instance (which is really a virtual machine) provided by Windows Azure. And although the figure shows only two instances, a client can start 10, 20, or 100 if needed.

Think what this allows: With just a single on-premises machine running Windows HPC Server, an organization can have on-demand access to a cluster. Whenever an application needs the cluster's power, an administrator can create the cloud instances and the user can run the application. When it's completed, the administrator can shut down the instances. The organization is charged only for the time the cloud instances are running.

It's worth pointing out that not every application will work well in this environment. MPI jobs, for example, require communication between compute nodes while they execute, and so clusters designed for this application style frequently have high-speed networks between those nodes. Windows Azure doesn't provide these today, which means that some kinds of MPI jobs might not perform well in the cloud. Embarrassingly parallel jobs can work well in this situation, however, as can Excel applications.

Given the small investment required—all that's needed is a head node—it's reasonable to expect many organizations to create this kind of on-demand cluster. Even a small company can now have access to large amounts of affordable computing power. For example, an application that's usually run on a single workstation can be sent to an on-demand cluster whenever its needs outstrip that machine's capacities. The ability to create a set of cloud instances when required provides a useful and low-risk way to get started with cluster computing.

## Creating Data-Intensive Applications

Cluster computing has traditionally been associated with compute-intensive applications. By definition, this class of software is CPU-bound; making an application run faster requires providing more processing power. But clusters can also be used to speed up execution of I/O-bound jobs.

In data-intensive scenarios, for example, applications need to read large amounts of unstructured, non-relational data. The processing done on this data is typically quite simple—lots of CPU power isn't required. Instead, the challenge is to read a large amount of information from disk as quickly as possible. For applications whose logic can process different parts of that data in parallel, a compute cluster can help.

A cluster can provide two distinct services for data-intensive applications:

- It can offer a relatively inexpensive place to store large amounts of unstructured information reliably.
- It can provide a framework for creating and running parallel applications that process this data.

Windows HPC Server 2008 R2 SP 2 supports both of these things, as Figure 3 illustrates.

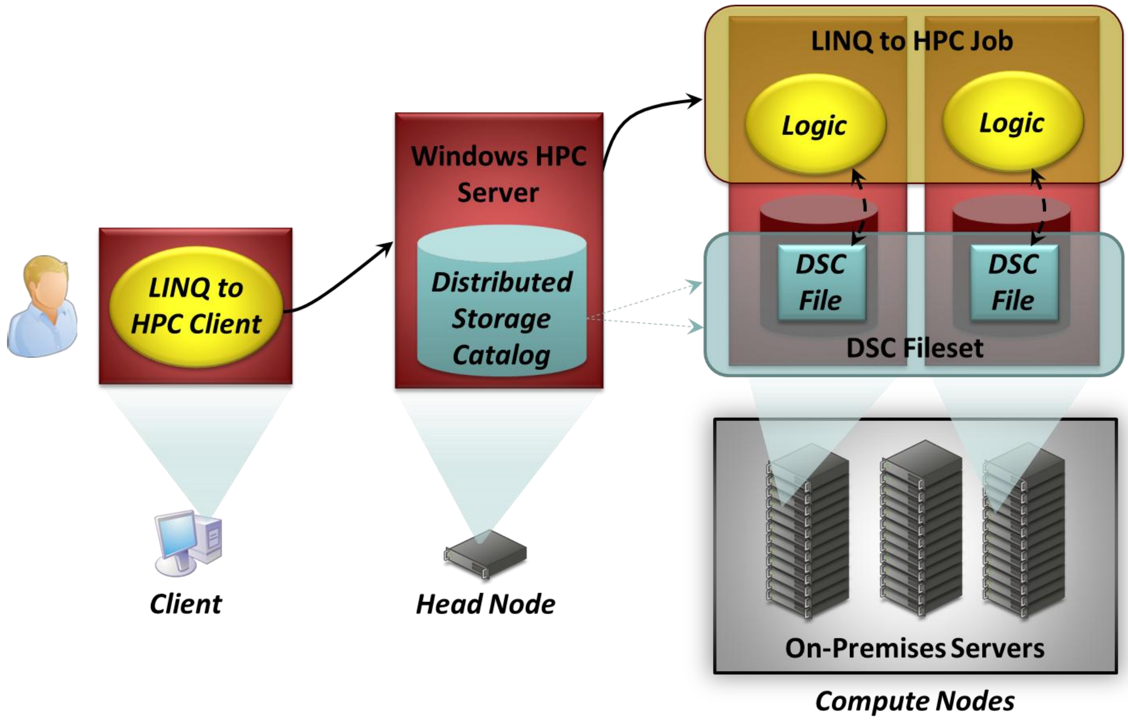


**Figure 3: A cluster can be used to store large amounts of unstructured data and to run applications that process that data.**

As the figure shows, a *Distributed Storage Catalog (DSC)* runs on the cluster's head node. The DSC keeps track of all of the *DSC files* that belong to a particular *DSC file set*. A DSC file is really just an ordinary Windows file, but because the DSC files are logically grouped into a DSC file set, an application running on the cluster can refer to all of them, no matter how large, as a single unit. And because each DSC file is replicated at least three times across the machines in the cluster, the DSC file set as a whole can be highly available. If a cluster is built using on-premises commodity servers, DSC file sets can be an inexpensive way to store large amounts of unstructured data.

To process this data, a user can create a *LINQ to HPC* application. A developer writes a *LINQ to HPC client*, which expresses operations on data using Microsoft's Language-INtegrated Query (LINQ) technology. Running this client generates a *LINQ to HPC job*, which gets submitted to the head node as usual and executed on the cluster. To make this job run as fast as possible, its logic is broken into chunks, with each chunk started on a machine that holds a copy of the DSC file that chunk will work on. This allows local access to data, letting each chunk read and process its piece of the DSC file set in parallel. The result is that an I/O-bound application can run faster.

The need for data-intensive applications is growing rapidly. Supporting these applications on a cluster makes sense, and it provides an important illustration of how modern clusters can be used.

## Conclusion

Because cluster computing can now support a broader range of applications, it's become more useful. Because clusters can now contain a range of compute nodes, including on-premises servers, desktop workstations, and cloud instances, cluster computing has become more accessible. For anybody who thought this approach to making software run fast wasn't relevant to them, it's time to take another look.

## About the Author

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.